ISSN:2790-1688 DOI: 10.56028/aetr.4.1.295.2023

A Survey of Blockchain-Based Smart Contract Application Testing Framework in the Energy Industry

Jingjian Chao[1*], Huixia Ding[1], Shuai Fang[1]

Ting Rui[1],Lining Zhang[1], Honglin Xue[2]

China Electric Power Research Institute Beijing,100192, China

Information and Communication Branch of State Grid Shanxi Electric Power Company

ShanXi, 030021, China

chaojingjian@yeah.net; dhx@epri.sgcc.com.cn;

fangshuai@epri.sgcc.com.cn;xintong-ruiting@epri.sgcc.com.cn;812925746@qq.com

Abstract. Blockchain is the fundamental component of smart contract applications. Testing technology plays a special and irreplaceable role in the development of smart contract applications in blockchain-based applications, especially in distributed renewable energy transaction scenarios in the energy industry. In practice, the formulation of blockchain technology as a standard infrastructure is an essential means to improve the reliability of blockchain-based applications in the energy industry. However, the quality of the organization-level blockchain still encounters many challenges, such as password attacks, and double spending attacks, which attract much attention from both research and academic area. Much research has focused on quality improvement through testing to fulfill the requirement toward functional, performance, and security requirements of the industry. However, the framework to accomplish the specific testing task was not present comprehensively yet. In This paper, an investigation was given on the supervision and testing of blockchain-based applications in the energy industry after its operation online. The existing testing indicator, model, and application scenario combined with the practice in the state grid industry were illustrated, in which the process, models, and methods were shown, and also suggestions were given on promoting the blockchain-based smart contract testing evaluation in the energy industry.

Keywords: Smart Contract, Testing, Energy, Survey.

1. Introduction

In recent years, blockchain received much attention from academics and the industry for its tamper-proof feature[1]. Lots of algorithms such as mathematics, cryptography, and computer programming languages were proposed to achieve this feature. The smart contract is one of the most important practices based on blockchain[2]. From the perspective of the application, blockchain can be deemed as a distributed and shared ledger or database, decentralized, non-tampered, and traceable throughout the whole life cycle. These characteristics ensure the trustable of the blockchain and construct the cornerstone of the blockchain application. The application of the blockchain basically relies on the fact that it can solve information asymmetry and assure trust during collaborative thus concerted action among multiple organizations.

However, many risks still exist in the current blockchain system, which includes consensus mechanism security risks, smart contract security risks, and encryption mechanism security risks[3]. Consensus mechanism security risks include internal and external attackers which can use the consensus mechanism's own design vulnerabilities, node failure or link breakage, false identity, and other vulnerabilities to destroy the consistency, reliability, and availability of the consensus mechanism, resulting in consensus failure and convergence time[4].

In most work on the risk elimination of blockchain-based applications, testing was a promising method to verify the performance efficiency[5], network communication mode, encryption module availability, consensus mechanism, node management, and many other evaluation indicators.

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.295.2023

Smart contract security risks mainly come from risks such as smart contract operating environment loopholes and smart contract own code and logic loopholes, including contract programming security loopholes[6], compiler errors, Ethereum virtual machine errors[7], etc. Attackers can use logical loopholes and code loopholes to implement operations that do not conform to the smart contract agreement.

Cryptographic mechanism security risks include key distribution management risks, cryptographic algorithm design backdoors[8], vulnerabilities during development, etc. In addition, with the increasing maturely application of quantum computing technology, it can be possible to crack asymmetric cryptographic algorithms in seconds.

Currently, the quality of the blockchain is the deficiency that hide the large-scale application in the industry, and security incidents occurs with great economic loss which sounded the alarm for the industry. To make the blockchain more efficient, it is necessary to accelerate the exploration of the functional verification framework of the blockchain, establish a quality assurance framework that adapts to the technical evolution of the blockchain, and ensure the security and credibility of the blockchain based application.

The rest of the paper is organized as follows. Section 2 shows a general picture of Blockchain supervision in Research and industry while Section 3 reviews the functional testing framework, performance testing framework, security testing framework, indicator, and password testing framework are discussed. In the end, Section 4 summarizes our conclusions and highlights future direction.

2. Related Work

Smart contract was first proposed and concretely described by Nick Szabo in 1994. In which smart contract was defined as a set of commitments written in digital form, including the protocol on which contract participants can execute these commitments[9]. The smart contract has greatly expanded practical scenarios of blockchain. However, frequent security incidents seriously hinder its development. The main reasons for the security problems are: 1) The credibility of the smart contract comes from its tamper resistance, which cannot be modified once it is deployed online. 2) Many source codes of smart contracts were disclosed, which can improve trust in the contract, however, it also greatly reduces the cost of hacker attacks. Smart contracts on the open network are becoming the target of professional hackers. 3) Potential deficiency embedding in the code of contract during the development process of smart contracts.

For the research on the security assurance of smart contracts, ten types of security problems were summarized that occur most frequently in smart contracts, respectively: code re-entry, access control, integer overflow, not strictly judging the return value of unsafe function calls, denial of service (DoS), predictable random processing, competitive conditions/illegal advance transactions timestamp dependency, short address attacks, and other vulnerability types[10].

According to the operation mechanism of smart contracts, the life cycle of which can be summarized as six phases: negotiate phases, develop phases, deploy phases, operation, maintenance phases, learning phases, and self-destruction phases. The development stage includes contract testing before the contract deploys on the chain. Smart contracts are divided into the infrastructure layer, contract layer, operation and maintenance layer, intelligent layer, presentation layer, and application layer[11]. Among them, the operation and maintenance layer encapsulates a series of dynamic operations on static contract data in the contract layer, including contract algorithm design, formal verification of contract, security check by security testing, maintenance update during operation, and self-destruction at the end of the smart contract life cycle. Smart contracts with security vulnerabilities will bring huge economic losses. The operation and maintenance layers are the keys to ensuring that smart contracts can operate safely.

At present, some security checking tools for contracts were proposed, such as Oyente and Mythril[12], [13]. Which draws the contract bytecode into a control flow diagram and analyses

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.295.2023
common security vulnerabilities. However, this method cannot verify the functional correctness of the contract. The security vulnerabilities that can be detected are limited and may cause false alarms. Bhargavan et al. proposed a verification framework for the functional correctness of the Ethereum Solidity contract. It converts Solidity language and EVM bytecode into F * language to verify various attributes of the code, which can both eliminate vulnerabilities and calculate the contract consumption gas limit[14]. Similar formal verification tools for smart contracts include ZEUS[15], Manticore[16], Solgraph[17], etc.

3. Overview of the framework of blockchain in the state grid

3.1 Functional Testing Framework

In the functional testing framework of blockchain in State Grid, the requirements are first formed into a system functional requirement specification, which defines the functional requirement of the blockchain system. The system function test cases of the blockchain system were developed according to the system function test instructions, and then test the system functions according to the test cases.

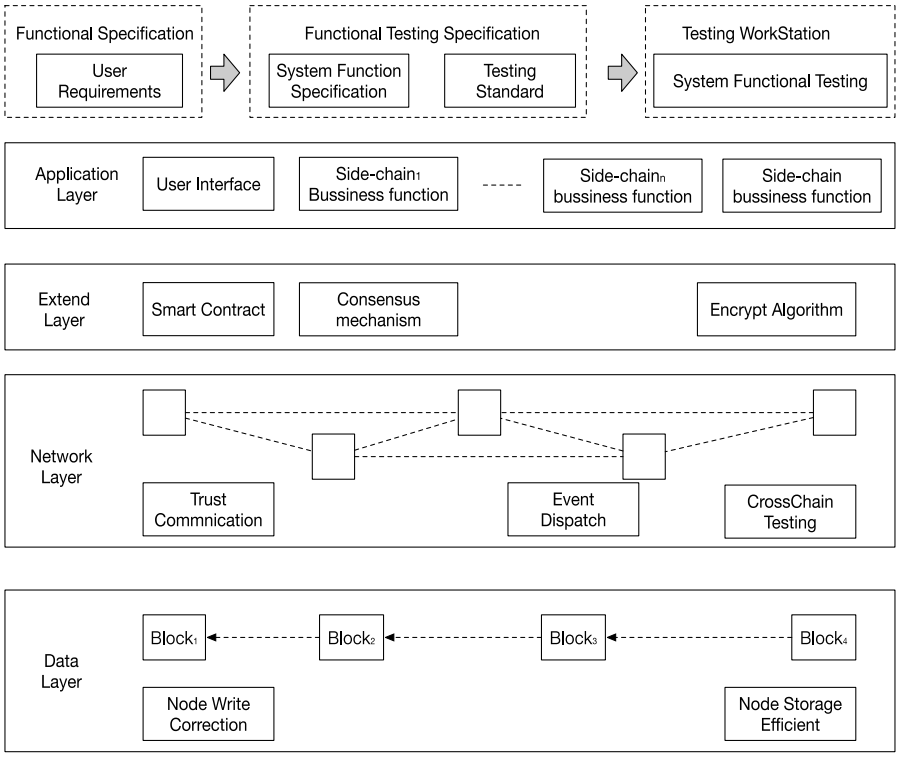

Figure 1 Functional Testing Framework of Block Chain in State Grid

The specific roadmap is shown in Figure 1. In the process of functional testing of the blockchain system, the system functions of nodes with specific functions in the main chain or different side chains are tested on the testing machine connected to the API interface of the main chain, transaction chain, and data chain. The different permissions of the attributes of the nodes complete the test of the test cases of the designed blockchain system functions.

3.2 Indicators and Method for Performance Testing Framework

The performance testing of the blockchain-based application in the state grid is illustrated in Figure 2. Foremost, the user implements the transaction request by calling and executing the corresponding smart contract after the transaction submitting and confirming. The changes in hardware environment resources such as memory, disks, and the number of transactions completed within a given time are used to measure the performance of the blockchain system. And constitute

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.295.2023
indicators of performance Testing. The system performance is tested mainly through three aspects: capacity, resource utilization, and temporal characteristics.

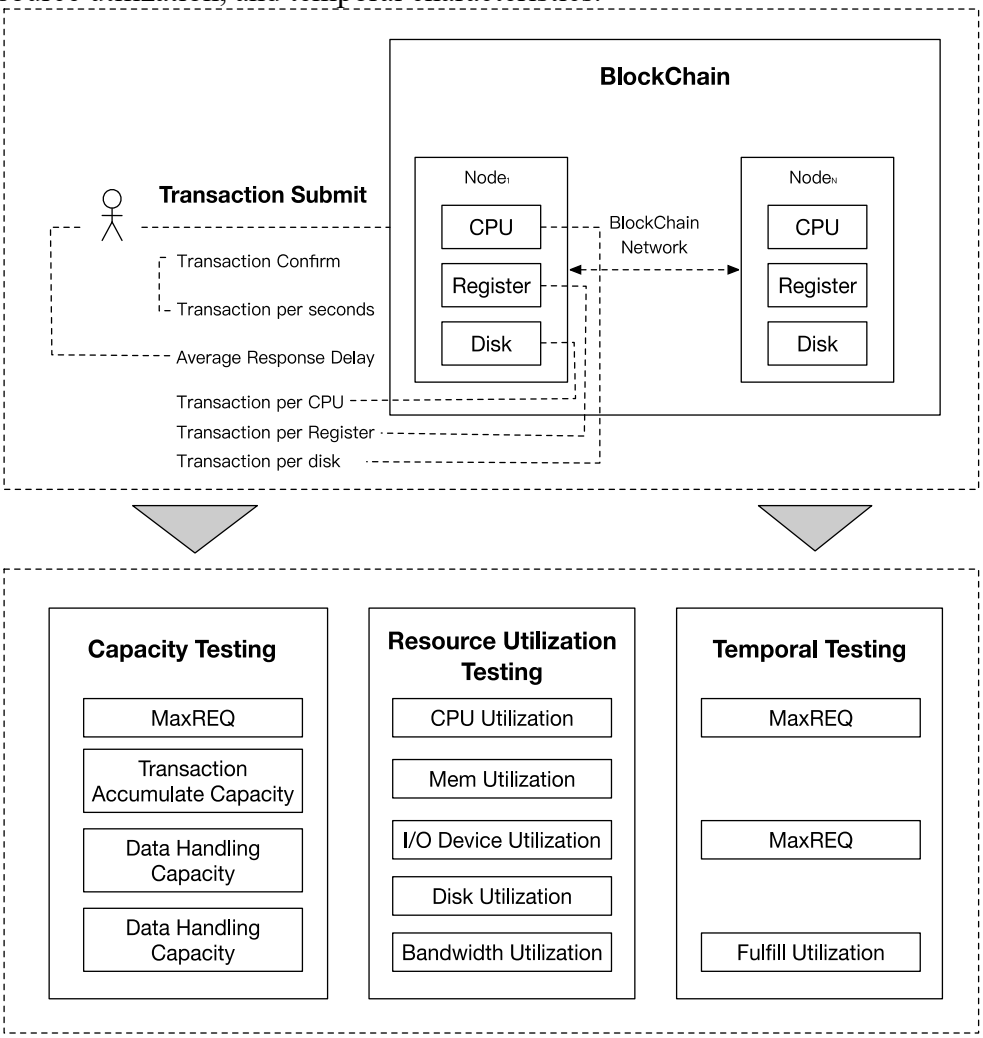

Figure 2 Performance Testing Framework of Block Chain in State Grid

The performance testing of the blockchain system relies on a unified hardware environment, a unified test tool, and a unified test standard. In the process of performance testing the blockchain system, the first step is to formulate standard and standardized test methods and use cases. According to the key performance indicators related to the blockchain system, as can be seen in Figure 2, the developed blockchain system performance test cases mainly include capacity testing, resource utilization testing, and temporal characteristic testing.

3.3 Indicators and Method for Security Testing Framework

The security test for the blockchain system is carried out after the blockchain system was constructed. As it is shown in Figure 3, security testing mainly for node management, visit control, identity management, consensus mechanism, smart contract, regulatory support, security operation and maintenance, security governance, etc. The purpose of the framework is to carry out comprehensive security testing to find the loopholes and security problems in the blockchain system to eliminate or alleviate the security threats of the blockchain-based applications.

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.295.2023

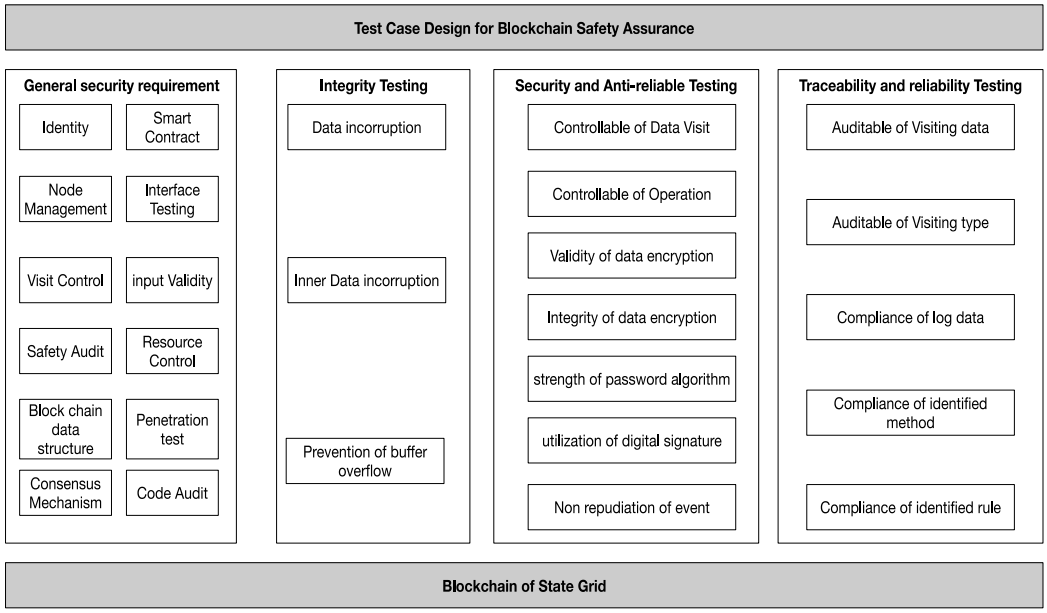

Figure 3 Security testing framework of blockchain in State Grid

As it is shown in Figure 3, Security testing mainly includes general security requirements such as confidentiality testing, confidentiality evaluation elements, integrity evaluation, non-repudiation evaluation, and traceability evaluation.

3.4 Indicators and Method for Password Testing Framework

For the test of cryptographic algorithms in the blockchain system, the cryptographic algorithms are classified according to their functions and characteristics, and the cryptographic algorithms are divided into three categories: cryptographic hash algorithm, encryption, and decryption algorithm, and signature algorithm as it is shown in Figure 4. According to these three types of cryptographic algorithms, the test case was executed to test and analyze the security and performance.

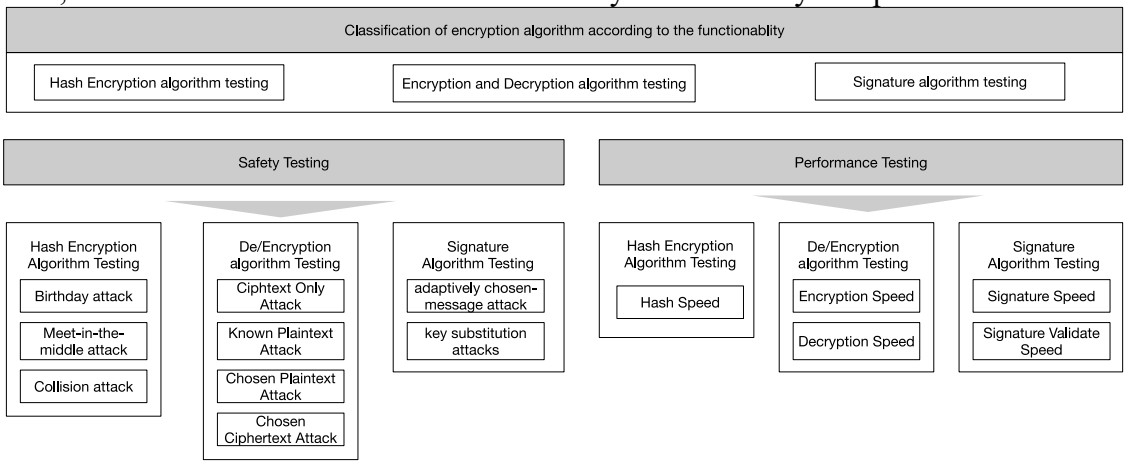

Figure 4 Password Algorithms testing framework of blockchain in State Grid

When conducting security testing of cryptographic algorithms, the test in blockchain systems is mainly carried out by simulating attacks. During the specific test process, the security of the password hash algorithm was tested by using some algorithms such as birthday attack algorithms, meet-in-the-middle attack algorithms, differential attack algorithms, and collision attack algorithms. When analyzing the security of the encryption and decryption in blockchain applications, the algorithm is tested by using the common attack testing methods such as ciphertext-only attack, known plaintext attack etc. When testing the security of the signature algorithm, security of the signature algorithm is tested by methods such as self-chosen message attacks and key replacement attacks. When performing the performance test of the cryptographic algorithm, it is mainly to test

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.295.2023
the hash value calculation speed of the cryptographic hash algorithm, including the encrypt and decrypt speed of the corresponding algorithm, and the signate speed and signature verification speed of the signature algorithm.

4. Conclusion

In this paper, a testing framework in the state grid of blockchain was introduced. Through the description of the blockchain test framework, the function test, performance test, security test, and password algorithm test of the blockchain system are implemented. During the specific test process, the testing can be carried out according to the test technical standards and test specification processes to achieve the control of defects, quality, and versions. Which can be a reference for the construction of a blockchain testing system with more comprehensive testing capabilities.

Acknowledgment

This work was supported by State Grid Technology Project Grant(5700-202158411A-0-0-00).

References

[1] F. Casino, T. K. Dasaklis, and C. Patsakis, “A systematic literature review of blockchain-based applications: Current status, classification and open issues,” Telematics and informatics, vol. 36, pp. 55–81, 2019.

[2] B. K. Mohanta, S. S. Panda, and D. Jena, “An overview of smart contract and use cases in blockchain technology,” in 2018 9th international conference on computing, communication and networking technologies (ICCCNT), 2018, pp. 1–4.

[3] J. Lindman, V. K. Tuunainen, and M. Rossi, “Opportunities and risks of Blockchain Technologies–a research agenda,” 2017.

[4] G. S. Aujla, A. Singh, M. Singh, S. Sharma, N. Kumar, and K.-K. R. Choo, “BloCkEd: Blockchain-based secure data processing framework in edge envisioned V2X environment,” IEEE Trans Veh Technol, vol. 69, no. 6, pp. 5850–5863, 2020.

[5] B. Koteska, E. Karafiloski, and A. Mishev, “Blockchain implementation quality challenges: a literature,” in SQAMIA 2017: 6th workshop of software quality, analysis, monitoring, improvement, and applications, 2017, vol. 1938, p. 8.

[6] R. M. Parizi, A. Dehghantanha, and others, “Smart contract programming languages on blockchains: An empirical evaluation of usability and security,” in International Conference on Blockchain, 2018, pp. 75–91.

[7] E. Hildenbrandt et al., “Kevm: A complete semantics of the ethereum virtual machine,” 2017.

[8] [8] J. Guo, “Risks of the blockchain technology,” in International conference on Big Data Analytics for Cyber-Physical-Systems, 2020, pp. 1903–1909.

[9] N. Szabo, “Formalizing and securing relationships on public networks,” First monday, 1997.

[10] M. A. Khan and K. Salah, “IoT security: Review, blockchain solutions, and open challenges,” Future generation computer systems, vol. 82, pp. 395–411, 2018.

[11] M. Kolvart, M. Poola, and A. Rull, “Smart contracts,” in The Future of Law and etechnologies, Springer, 2016, pp. 133–147.

[12] P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Buenzli, and M. Vechev, “Securify: Practical security analysis of smart contracts,” in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 67–82.

[13] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, “Making smart contracts smarter,” in Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, 2016, pp. 254–269.

ISSN:2790-1688 DOI: 10.56028/aetr.4.1.295.2023
[14] K. Bhargavan et al., “Formal verification of smart contracts: Short paper,” in Proceedings of the 2016 ACM workshop on programming languages and analysis for security, 2016, pp. 91–96.

[15] S. Kalra, S. Goel, M. Dhawan, and S. Sharma, “Zeus: analyzing safety of smart contracts.,” in Ndss, 2018, pp. 1–12.

[16] M. Mossberg et al., “Manticore: A user-friendly symbolic execution framework for binaries and smart contracts,” in 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2019, pp. 1186–1189.

[17] E. Zhou et al., “Security assurance for smart contract,” in 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2018, pp. 1–5..